UNIVERSITÀ DI PISA

COMPUTER ENGINEERING

Foundations of Cybersecurity

A.A. 2023-2024

Bulletin Board System

Francesco Taverna

INDEX

1 INTRODUCTION

This project consists of a cryptographically secure message application, developed in C++ for Linux systems.
The application is made up of a BBS Server, which impersonates the message container (called BBS), and one or more clients that contact it via TCP protocol to perform 3 functions made available:

- Register to the Server
- Login to the Server
- Add message
- Read the last T messages available

The project has been divided into subdirectories for readability and convenience.
In communication between client and server, the length of the packet is sent first and then the packet is sent, to manage the length of the packets correctly.

1.1 SERVER

The “Server” subdirectory contains the server Main (MainServer.cpp) and the file containing all the Server functions (Server.cpp).
The purpose of the server is to listen and serve the clients, providing the functions of the BBS application (Register, Login, Add message, Read the last T messages available).

1.2 CLIENT

The 'Client' subdirectory contains the client-related main (MainClient.cpp) and the various client-related functions (Client.cpp).
The client must first authenticate the server via the Diffie Hellman protocol, then can register or log in with Nickname and Password and can thus perform the functions made available to the server.

1.3 DTO

The “DTO” (Data Transfer Object) subdirectory is a directory that contains the *Message* and *User* classes, classes useful in managing the user and the messages carried out by the client.

1.4 FILEMANAGER

The “FileManager” subdirectory is made for finding useful functions for managing files. In particular, functions that allow us to check the password, saved hashed in a specific file, to insert messages in the *BBS_messages.txt* or to read the last messages from the latter.

1.5 PACKETS

The “Packets” subdirectory contains the 3 types of packages used in the project:

- StartPacket
- AuthenticationPacket
- GenericPacket

The StartPacket is used by the Client to initiate the connection with the server.
The AuthenticationPacket is used by both the client and the server to authenticate each other.
The GenericPacket is used as a generic message between server and client after successful authentication.

1.6 UTIL

The “Util” subdirectory contains two files that are absolutely essential for the project.
The first is *UtilFunctions.cpp*, which contains various functions used throughout the project.
In addition, *CryptoFunctions.cpp* contains all the cryptography-related functions used within the project, which allowed us to make the banking application cryptographically secure.

1.7 USERS

The “Users” subdirectory contains 2 types of files for each user:

- nickname.txt
- password.txt

The *nickname.txt* file contains various information related to the username, such as: nickname, salt and counter.
The *password.txt* file contains the password salted and hashed, also for security reasons.

2 PROTOCOLS

To develop this project and implement the cryptographic functions it has been used the OpenSSL 3.0 library.
It has been generated a pair of keys for the server. These keys were created using the openssl library issuing the following commands from the Linux terminal:

- *openssl genpkey -algorithm RSA -aes256 -out private_key.pem*
- *openssl rsa -pubout -in private_key.pem -out public_key.pem*

To perform the initial authentication between server and client has been used the Diffie Hellman protocol thanks to which two shared secrets were derived, one used as a symmetric key for communications, using the AES256 protocol, and one to generate the HMAC digest using the SHA512 protocol.
The packets exchanged after authentication are therefore encrypted with the key to ensure **confidentiality**.
A signed HMAC digest is added to the package to ensure **integrity** and **authenticity**.
A counter has been added to guarantee **freshness**, to avoid replay attacks.
User passwords are saved salted and hashed, to avoid saving keys in clear and adding randomness to the hash.

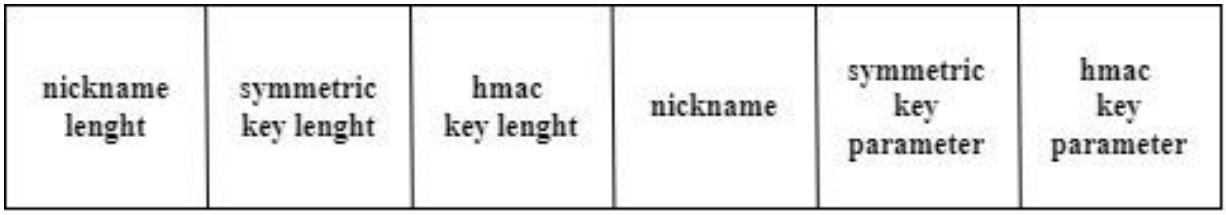
NOTE: The keys sent to establish the shared secret are serialized and deserialized using the BIO structures provided by the openssl library.

3 PACKETS FORMAT

3 different types of packages have been created to handle communication between client and server.

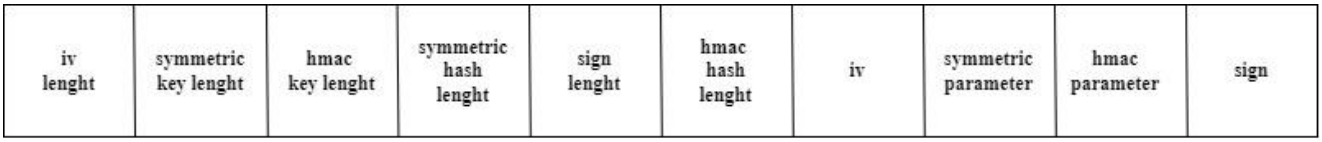
3.1 STARTPACKET

The StartPacket is used by the client to begin the authentication process on the server, both for registration and login. The client generates the two Diffie-Hellmann parameters (symmetric and hmac keys) and sends nickname length, parameter length, nickname (empty if it's registration phase) and parameters in clear.



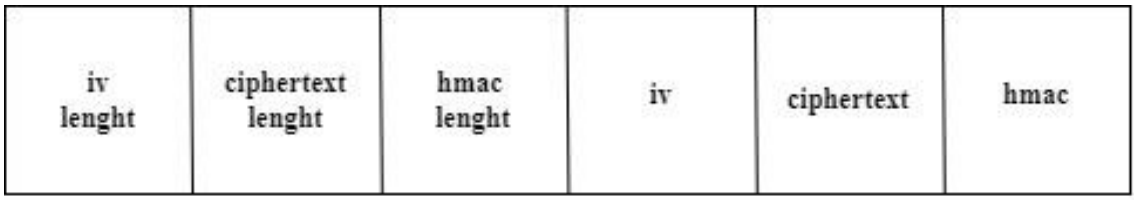
3.2 AUTHENTICATIONPACKET

The AuthenticationPacket is sent by server to client in order to certify to the latter that he is talking with real entity. The server uses this packet to send its network parameters in the clear, so that the client can also construct the symmetric keys.
The signature is also sent, which is formed by the concatenation of the two keys first hashed and then signed, so as to guarantee **authentication** and **integrity**.



3.3 GENERICPACKET

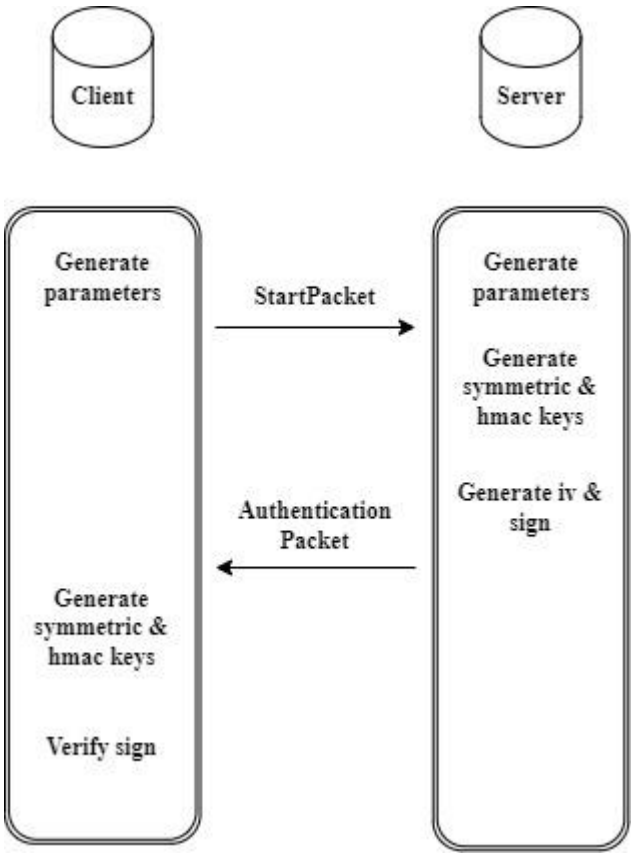
The GenericPacket is used as a generic communication packet between client and server to manage the various functions offered by the server.
The packet is composed of the *iv*, used to encrypt and decrypt, the *ciphertext* encrypted with the *symmetric key* and the *hmac*, used to guarantee integrity, encrypted with the hmac key.



4 WORKFLOW

4.1 AUTHENTICATION

For the authentication between client and server, the workflow is managed using the StartPacket and AuthenticationPacket.



4.2 COMMUNICATION

Generic packets are used for communication between server and client in a symmetric key.